

APPLICATION FOR UNITED STATES LETTERS PATENT

INVENTORS: Jae Ho Jung

TITLE: METHOD OF STORING DATA IN A PERSONAL
INFORMATION TERMINAL

ATTORNEYS: FLESHNER & KIM, LLP
& P. O. Box 221200
ADDRESS: Chantilly, VA 20153-1200

DOCKET NO.: LT-0010

METHOD OF STORING DATA IN A PERSONAL INFORMATION TERMINAL

BACKGROUND OF THE INVENTION

1. Field of the Invention

[1] The present invention relates to a Personal Data Assistant (PDA), and more particularly to a method of storing data in a plurality of application programs embedded in a PDA.

2. Background of the Related Art

[2] A Personal Data Assistant (PDA) or Personal Information Manager (PIM) is a small portable information terminal having many functions of a computer and an electronic organizer, and may also have communication functions. A PDA typically uses a touch screen as an input/output device. The touch screen displays icons associated with various embedded application programs, as shown in Figure 1.

[3] A PDA may have various basic application programs embedded in it. Examples of the basic embedded application programs include a "Calendar" for scheduling appointments and meetings, "Contacts" for tracking friends and colleagues, "Tasks" for keeping a to-do list, and "Notes" for recording thoughts and ideas.

[4] To use the PDA, a PDA user chooses an application program suitable to data that the user wishes to enter and store, and executes the application by tapping a

corresponding icon displayed on the touch screen. When the application is opened, the user enters desired data in the PDA through the executed application program. The entered data is formatted in a way that is suitable for the particular application program, and is stored in a database (DB) corresponding to the application.

[5] For example, if a PDA user wants to record his or her ideas or thoughts, or wants to search for a previously stored note, the user taps a “Notes” icon displayed on the touch screen to execute the basic Notes program. After execution of the Notes program, the user would enter his or her ideas or thoughts, or review a previously stored note.

[6] In addition, if the user wants to enter a private schedule event consisting of date, time, location, and agenda, or to review a previously stored event, the user would tap a “Calendar” icon to execute the basic Calendar program. The user could then enter a private schedule event or confirm a previously stored schedule event.

[7] Further examples of the operations of the aforementioned basic application programs are described in chapter 4, “Microsoft Pocket Outlook”, of the manual entitled “iPAQ H3000 Pocket PC Reference Guide” for a PDA manufactured by Compaq.

[8] Each of the aforementioned basic application programs has its own DB for individually storing entered user data. Each DB has fields different from the other DBs. For example, the “Calendar” program accepts fields such as time, date, and so forth. On the other hand, the “Contact” program accepts fields such as name, address, telephone

number, and so forth. Therefore, each application program has an individual DB having different fields.

[9] Because of such field differences among individual DBs, if an entry needs to be entered in the “Contact” program where the desired data is being entered through the “Notes” program, the “Contact” program must be tapped and executed, and the data needs to be re-entered and stored. Additionally, if a private schedule entry is to be entered as well, the “Calendar” program must be executed and the desired schedule entry is entered into a corresponding DB through the “Calendar” program.

[10] Thus, if a user writes data into the “Notes” program that is suitable for the “Contacts” program (for example telephone number and address), and also writes data that is suitable for the “Calendar” program, (for example, appointment time, schedule event, location, and agenda) all of the data must be re-entered after executing an appropriate application program. This repetition of data entry and program execution creates a duplication of effort and can be tedious and inconvenient.

[11] The above references are incorporated by reference herein where appropriate for appropriate teachings of additional or alternative details, features and/or technical background.

SUMMARY OF THE INVENTION

[12] An object of the invention is to solve at least the above problems and/or disadvantages and to provide at least the advantages described hereinafter.

[13] It is another object of the present invention to provide method of storing data in a PDA terminal, which can convert a format of data entered for an arbitrary executed application program to other format suitable for another application program.

[14] It is another object of the present invention to provide a method of storing data in a PDA, which can store format-converted data in the another application program without data re-entering or additional program execution.

[15] In order to achieve at least the above objects in whole or in parts, there is provided a method of storing data, including composing data in one of a plurality of application programs embedded in the personal data assistant, the application programs having mutually different data formats; converting a format of the composed data to a format suitable for a second application program with or without assistance of a user; and storing the format-converted data in the second application program database.

[16] In order to further achieve at least the above objects in whole or in parts, there is provided a method of storing data in a personal information terminal, including composing data having a prescribed identifier code in a first one of a plurality of application programs, the prescribed identifier code being indicative of a second one of the plurality of application program into which the composed data is to be stored,

selecting the second application program among the plurality of application programs based on the prescribed identifier code using a table to match prescribed identifier codes to corresponding application programs, converting a format of the composed data from a first format to a second format, the second format corresponding to a format required by the second application program, and storing the data in a database associated with the second application program.

[17] In order to further achieve at least the above objects in whole or in parts, there is provided a method of storing data in a personal information terminal, including composing data of a first format in a first one of a plurality of application programs embedded in the personal information terminal, each of the plurality of application programs having mutually different data formats, selecting a second application program in which to store the composed data, converting a format of the composed data to a second format used by the second application program, and storing the format-converted data in the second application program.

[18] In order to further achieve at least the above objects in whole or in parts, there is provided a personal data assistant (PDA), including an input/output interface, configured to receive commands from a user and display information, a central processing unit (CPU), configured to receive and process commands entered into the PDA, and a memory, configured to accommodate a plurality of data bases associated with a plurality of application programs, wherein a first application program stored on the PIM is

configured to receive and store data in a first database using a first data format, a second application program stored on the PDA is configured to receive and store data in a second database using a second data format, and wherein a conversion program is executed by the CPU to receive the data in the first data format, reformat the received data to the second data format, and store the reformatted data in the second database.

[19] In order to further achieve at least the above objects in whole or in parts, there is provided a computer readable medium having stored thereon a sequence of instructions which, when executed by a processor, cause the processor to perform a sequence of steps, including reading data of a first format from a first application program, converting the first format to a second format, the second format being suitable for a second application program, storing the data in the second format in a database associated with the second application program.

[20] Additional advantages, objects, and features of the invention will be set forth in part in the description which follows and in part will become apparent to those having ordinary skill in the art upon examination of the following or may be learned from practice of the invention. The objects and advantages of the invention may be realized and attained as particularly pointed out in the appended claims.

BRIEF DESCRIPTION OF THE DRAWINGS

[21] The invention will be described in detail with reference to the following drawings in which like reference numerals refer to like elements wherein:

[22] Figure 1 illustrates a related art personal data assistant (PDA);

[23] Figure 2 is a block diagram of a PDA according to a preferred embodiment of the present invention;

[24] Figure 3 is a schematic diagram illustrating various application programs to be linked with a format converting program according to the preferred embodiment of the present invention;

[25] Figure 4 is a flow chart illustrating a method of storing data according to the preferred embodiment of the present invention;

[26] Figures 5 to 7 are exemplary PDA screens displayed by a data storing method according to the preferred embodiment of the present invention; and

[27] Figure 8 is a flow chart of another preferable embodiment of a data storing method according to the preferred embodiment of the present invention.

DETAILED DESCRIPTION OF PREFERRED EMBODIMENTS

[28] Figure 2 is a block diagram of a PDA using a data storing method according to the preferred embodiment of the present invention. The PDA preferably includes a

touch screen 11 to display a plurality of icons associated with various embedded application programs and to detect a user's tapping, and a key pad 12 to receive key inputs. The PDA also preferably includes various application programs (APPs) 20₁ - 20_n, such as "Notes" and "Calendar", each having its own DB for storing received data after appropriate formatting.

[29] Next, a format converting program 30 is provided to convert a format of data received from an arbitrary APP 20₁ - 20_n to a format suitable for another APP, and an operating system (O/S) 15 is provided to enable communication channels among the APPs 20_x and to drive hardware resources. Finally, the PDA includes a CPU 10 to execute an APP selected by a user and manage all of the elements to conduct an operation requested by a user, through the assistance of the O/S 15.

[30] Referring to Figure 3, the format converting program 30 preferably conducts the conversion of data between the various formats, such as "Notes" 20₁, "Calendar" 20₂, and "Contacts" 20₃, etc. For example, while a user is entering desired data through the "Notes" program 20₁, the format converting program 30 preferably convert the entered data to another application format. Thus, unformatted private schedule data can be converted to a format that is acceptable to the "Calendar" program 20₂. The data storing method is described below in further detail.

[31] Figure 4 is a flow chart showing a data storing method according to the preferred embodiment. Referring to Figure 4, it is assumed that several icons related with

various APPs $20_1 \sim 20_n$ have been displayed on the touch screen 11. If a user taps an arbitrary program, e.g., the “Notes” program 20_1 on the touch screen 11, the CPU 11 recognizes the selection of the “Notes” program 20_1 through the O/S 15 and executes the “Notes” program 20_1 (S10).

[32] Accordingly, the user may then freely enter desired data in an input screen of the executed “Notes” program 20_1 . This can be done through the touch screen 11 or any other data input method. The “Notes” program 20_1 adapts the entered data to its DB structure and stores the adapted data in its DB (S11).

[33] If the user wants the contents of the data entered in the “Notes” program 20_1 also to be stored in another APP, for example, the “Calendar” program 20_2 , the user preferably requests a format conversion by selecting a “conversion/store” menu item which may be prepared in all APPs (S11-1). The CPU 10 is preferably notified of the format conversion request through the O/S 15 when the “Notes” program 20_1 detects the selection of its “conversion/store” menu item. The format converting program 30 is then executed (S12). Alternatively, the user may set up the PDA to automatically convert entries without having to select the “conversion/store” menu item.

[34] The format converting program 30 preferably generates a menu screen, as shown in Figure 5, to receive a user’s selection of a target APP in which the stored data from the “Notes” program is to be stored after appropriate format conversion (S13). The menu screen may contain icons 51a, 51b associated with target APP candidates, e.g.,

“Contacts” and “Calendar”, which will receive the same contents of the data entered into the “Notes” program.

[35] Alternatively, instead of the candidate icons, the target APP candidates may be contained in a pop-up window which could be opened at the right-click of a mouse or selection of a menu. Many other methods are possible for selecting the target APP.

[36] When the user chooses a target APP, for example, the “Calendar” program, through the appropriate selection method (S14), the format converting program 30 preferably presents an information input window 52 in the touch screen 11. The information input window 52 preferably includes sub-windows corresponding to data fields necessary for storing data in a format suitable for the selected program (S15). The information input window to be activated for a given target APP is preferably pre-assigned.

[37] By way of example, the information input window for the “Calendar” program may contain fields of “date,” “time,” “location,” and “agenda.” Data entered in each field of the information input window is stored in a DB of the “Calendar” program by the format converting program 30 (S17).

[38] The format converting program 30 preferably stores the data entered in the corresponding fields in the “Calendar” DB. This can be done either with or without an intervention of the “Calendar” program. When the intervention of the “Calendar” program is provided, the format converting program 30 activates the “Calendar” program

first, and then, delivers the data entered in every fields through the communication channel of the O/S 15 to the activated “Calendar” program. The “Calendar” program will then store the received data in its DB.

[39] In the meantime, the CPU 10 continues monitoring whether termination of the running “Notes” program is requested from the user. If requested, it conducts a termination process of the “Notes” program.

[40] The above example described a process of converting information from the “Notes” program to the “Calendar” program. It should be understood that the process can be used to convert a data format from any first program to any second program.

[41] Additionally, in the above embodiment, data must be entered in prescribed fields to be stored in another APP. However, in a second embodiment of the present invention, data from a first APP can be automatically stored in a second APP without entering data in data fields. In this embodiment, for example, a user enters text or a memorandum in the “Notes” program in such a manner that the written order of its constituting words or phrases is matched with a field order of the target APP. Additionally, the user can set a block, if necessary, in the desired text or memorandum to be segmented into each data field of the target APP.

[42] For exact automatic segmentation, a user preferably enters delimiters in a text or a memorandum. For example, the user may enter delimiters before or behind each word or phrase corresponding to a respective data field, such as “date,” “time,” “location,”

“agenda”, and so forth, as shown in Figure 6. It should be understood that any delimiter, including a blank space or carriage return, could be used.

[43] Thus, according to the second embodiment, a user executes a first application by selecting the associated icon. For example, the “Notes” program is executed by tapping the “Notes” icon. Text or a memorandum can then be entered in accordance with a prescribed input rule. In the prescribed input rule, each tag is pre-defined. For example, the “\$C” tag could be used to indicate the “Calendar” program, the “\$T” could be indicative of the “Contacts”, and so forth. In addition, a specific symbol such as space, “/”, and “^” can be used as a delimiter.

[44] Thus, when a user enters a schedule entry in the “Notes” program according to the input rule, the user would write “\$C” first and then date, time, and agenda in order, with a delimiter between each data field.

[45] Figure 7 shows an exemplary memorandum regarding a meeting schedule entry adapted to the input rule. In the memorandum of Figure 7, the “\$C” tag is used to inform the format converting program 30 that the memorandum is associated with a schedule event, and the delimiter “^” is used. Additionally, “%C” is used to indicate the end of the schedule information, and “\$T”/“%T” indicates contact information.

[46] Figure 8 is a flow chart depicting an automatic format converting method of the format converting program 30 according to the second embodiment. First, a user initiates a first application program (S20). For purpose of example, the “Notes”

application will be described. Next, data is entered into the application (S21). The data should include the tags and delimiters as shown in Figure 7. Next it is determined whether a conversion/store command has been received (S22). If the user commands data conversion/store after entering text or a memorandum with the proper tags and delimiters, the format converting program 30 scans the entered text or memorandum copied from the “Notes” program through the communication channel of the O/S 15 to detect which tag has been included.

[47] If a tag is detected (S23), the format converting program 30 examines a prescribed tag-to-APP table to determine which APP is targeted for the entered text or memorandum (S24). If the detected tag indicates that the “Calendar” application is the target, the format converting program 30 recognizes that the entered text or memorandum is also to be stored in the “Calendar” program. The format converting program 30 thus segments the text or memorandum, excluding the tag, into each data field every time a delimiter is encountered (S25). The segmented data is then stored in the corresponding each field data entry of the DB of the targeted APP (S26).

[48] The tag, the delimiter, and field entry order may be changed freely by a user for the purpose of user-friendly data entering method.

[49] Each APP can store the entered data in its own DB allocated in program itself or in a file assigned to its DB.

[50] The method of storing data according to the disclosed embodiments has many advantages. For example, it provides for to a method for simply converting input data, entered in a format suitable for one of a plurality of application programs embedded in a personal data assistant (PDA), to have different format suitable for another application program chosen by a user, and storing the format-converted data in said another application program.

[51] The foregoing embodiments and advantages are merely exemplary and are not to be construed as limiting the present invention. The present teaching can be readily applied to other types of apparatuses. The description of the present invention is intended to be illustrative, and not to limit the scope of the claims. Many alternatives, modifications, and variations will be apparent to those skilled in the art. In the claims, means-plus-function clauses are intended to cover the structures described herein as performing the recited function and not only structural equivalents but also equivalent structures.